REMARKS/ARGUMENTS

Claims 1-4, 6-11, 13, 15-18, 20-21, and 23 are pending in the present application. In this amendment, Applicants have amended claims 1-2, 4, 18, 21, and 23, and canceled claim 5. Applicants are not conceding that the subject matter encompassed by claims 1-2, 4-5, 18, 21, and 23, prior to this Amendment, is not patentable over the art cited by the Examiner. Claims 1-2, 4, 18, 21, and 23 were amended, and claim 5 was canceled in this Amendment solely to facilitate expeditious prosecution of the application. Applicants respectfully reserve the right to pursue claims, including the subject matter encompassed by claims 1-2, 4-5, 18, 21, and 23, as presented prior to this Amendment and additional claims in one or more continuing applications.

I. 35 U.S.C. § 103, Obviousness

The Examiner has rejected claims 1-11, 13, 15-18, 20-21, and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over by Baba et al. (Pub. No. US 2002/0010733 Al) (hereinafter "*Baba*") in view of Hussain et al. (Pat. No. US 6,658,416 Bl) (hereinafter "*Hussain*") and further in view of Grunwald et al. (Whole-Program Optimization for Time and Space Efficient Threads, 1996 ACM) (hereinafter "*Grunwald*" - - art of record). This rejection is respectfully traversed.

Applicants' claim 1 recites, in part: "setting a thread tracking control bit in a processor of the data processing system to enable thread tracking by the processor" and "detecting, by the processor, a particular type of operation responsive to setting the control bit". The combination of the cited prior art does not render Applicants' claim 1 unpatentable because the combination does not teach or suggest these feature.

Baba teaches a processor that includes registers that are assigned to store pointers that indicate the status of a program, including the thread currently being executed and the location that has been reached in the thread's program. *Baba* does not teach or suggest, however, setting a thread tracking control bit in the processor and then detecting, by the processor, a particular type of operation responsive to setting the control bit.

Hussain teaches a program counter register that stores the program counter, but does not teach a control bit. In fact, Applicants could not find any reference to a "bit" in *Hussain* at all. Therefore, *Hussain* also does not teach or suggest setting a thread tracking control bit in the

processor and then detecting, by the processor, a particular type of operation responsive to setting the control bit.

Grunwald teaches saving and restoring registers during context switching. *Grunwald* also teaches performance counters, but does not teach or suggest a thread tracking control bit and then detecting a particular type of operation responsive to setting the control bit.

Applicants' claim 1 also recites, in part: "detecting, by the processor, a particular type of operation responsive to setting the control bit" and "responsive to detecting, and before executing, the particular type of operation: determining, by a trace application, whether an overflow is about to occur". The combination of the cited prior art does not render Applicants' claim 1 unpatentable because the combination does not teach or suggest these features or the interrelation of these steps.

The Examiner states that "Baba and Hussain do not explicitly disclose determining whether an overflow is about to occur in the work area; responsive to determining that an overflow is about to occur in the work area, copying the thread tracking information from the work area to a buffer". Office Action mailed May 2, 2008, page 5. The Examiner then relies on *Grunwald* to cure this deficiency of the combination of *Baba* and *Hussain*. Applicants disagree that the combination of *Baba*, *Hussain*, and *Grunwald* teaches the features of claim 1.

Grunwald teaches MMU hardware detecting an overflow. *Grunwald* also teaches checking that sufficient space remains in a given region in each procedure call. The check can occur prior to the procedure.

Grunwald does not cure the deficiencies of the combination of *Baba* and *Hussain* because *Grunwald* does not teach or suggest "detecting, by the processor, a particular type of operation responsive to setting the control bit". *Grunwald* does not teach or suggest detecting a procedure call responsive to setting a control bit. *Grunwald* simply teaches that procedure calls can be made, but does not teach or suggest detecting a procedure call responsive to setting a control bit.

Further, *Grunwald* does not teach or suggest "responsive to detecting, and before executing, the particular type of operation: determining, by a trace application, whether an overflow is about to occur". *Grunwald* teaches checking that sufficient space remains for the current activation record in each procedure call. *Grunwald* also describes the control over the thread stack space being possible with the assistance of a runtime library. *Grunwald* does not,

however, describe the runtime library being a "trace application" as required by Applicants' claim 1.

Applicants also claim determining, by a trace application, whether an overflow is about to occur in the work area using the current pointer register, the work area length register, and a total amount of data in the operation. The current pointer register includes a pointer pointing to a location of the work area where last thread tracking information is written. The combination of the cited prior art does not render Applicants' claim 1 unpatentable because the combination does not teach or suggest these features.

On page 6 of the Office Action mailed May 2, 2008, regarding original claim 2, the Examiner asserts that *Baba* teaches a work area register, a work area length register, and a current pointer register. On page 8 of the Office Action regarding claim 5, the Examiner asserts that *Baba* teaches "wherein the current pointer register includes a pointer pointing to a location of the work area where last thread tracking information is written". Specifically, the Examiner refers to the control register group and a current register data indicating the current state data of execution of a program.

As noted above, the Examiner asserts that "Baba and Hussain do not explicitly disclose determining whether an overflow is about to occur in the work area; responsive to determining that an overflow is about to occur in the work area, copying the thread tracking information from the work area to a buffer". Office Action mailed May 2, 2008, page 5. The Examiner then relies on *Grunwald* to cure this deficiency of the combination of *Baba* and *Hussain*.

The combination of *Baba*, *Hussain*, and *Grunwald* does not teach or suggest determining, by a trace application, whether an overflow is about to occur in the work area using the current pointer register, the work area length register, and a total amount of data in the operation. *Baba* teaches a control register group. The control register group includes registers in which may be stored pointers that indicate the status of a program. However, these registers are not used to determine whether an overflow is about to occur. As the Examiner notes, the combination of *Baba* and *Hussain* does not teach or suggest determining whether an overflow is about to occur.

The Examiner asserts that *Baba* teaches current register data indicating the current state data of execution of a program and that this teaches a "current pointer register" and the current pointer register including a pointer pointing to a location of the work area where last thread tracking information is written. Applicants respectfully disagree. Indicating the current state

data of execution does not teach a particular location. Applicants claim pointing to a location, which is narrower than merely teaching current state data in general. Therefore, *Baba* does not teach or suggest a current pointer register.

Further, the combination of *Baba*, *Hussain*, and *Grunwald* does not teach or suggest determining, by a trace application, whether an overflow is about to occur in the work area using the current pointer register, the work area length register, and a total amount of data in the operation. *Grunwald* teaches checking that sufficient space remains in a given region, but does not teach or suggest using registers to do this. In addition, checking that sufficient space remains in a given region does not teach or suggest using two registers, i.e. the current pointer register and the work area length register, as well as a total amount of data in the operation, where the current pointer register includes a pointer pointing to a location of the work area where last thread tracking information is written to make this check.

Applicants also claim "responsive to the trace application determining that an overflow is about to occur in the work area, sending an interrupt to an operating system to cause the operating system to copy the thread tracking information from the work area to a buffer using the set of hardware registers". The Examiner states that the combination of *Baba* and *Hussain* does not teach "responsive to determining that an overflow is about to occur in the work area, copying the thread tracking information from the work area to a buffer". The Examiner relies on *Grunwald* to teach this feature. Specifically, the Examiner asserts that *Grunwald* teaches allocating new stack space when there is a potential for overflow.

Grunwald does not cure the deficiencies of the combination of *Baba* and *Hussain*, however, because *Grunwald* does not teach or suggest sending an interrupt to an operating system to cause the operating system to copy the thread tracking information. Allocating new stack space does not teach sending an interrupt to an operating system.

Independent claims 18 and 21 have been amended in a similar manner as claim 1, and patentably distinguish over the cited art for similar reasons as discussed above with respect to claim 1.

The remaining claims depend from and further restrict one of the independent claims and also patentably distinguish over the cited art, at least by virtue of their dependency. Furthermore, many of these claims recite additional features which are not disclosed or suggested by the cited art.

For example, claim 2 depends from claim 1 and recites that the set of hardware registers include "a work area register." Claim 4 depends from claim 1 and recites that the work area register includes "one of a size of the work area for the thread or a pointer pointing to an end of the work area for the thread". In rejecting these claims, the Examiner refers to paragraph [0029] of Baba reproduced below, and to the register for stack head pointer, the register for program pointer and the register for local variable pointer, illustrated, for example, in Figures 4 and 6 of Baba:

> **[0029]** The stack machine controller is built into the aforesaid control mechanism. Any registers in the processor may be assigned as the control register group. The stack machine rewrites the control register group as needed as it accesses the work area for the current thread according to the data in the register group; in this way it executes the thread.

Although Baba may disclose various registers for accessing a work area, the reference does not disclose or suggest, either in paragraph [0029] of Baba or elsewhere, the specific registers recited in claims 2 and 4. Claims 2 and 4, accordingly, patentably distinguish over the cited prior art in their own right as well as by virtue of their dependency.

Furthermore, claim 13 depends from claim 1 and specifies that the buffer recited in claim 1 is one of a trace buffer and a consolidated buffer accessible by the application. Since, as discussed above, the references do not disclose a buffer to which tracking information is copied responsive to determining that an overflow is about to occur in a work area, the references also do not disclose or suggest that such a buffer may be a trace buffer or a consolidated buffer accessible by an application. Claim 13, accordingly, also patentably distinguishes over the cited art in its present form.

Therefore, the rejection of the claims under 35 U.S.C. § 103 has been overcome.

II. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: April 17, 2009

Respectfully submitted,

/Lisa L.B. Yociss/

Lisa L.B. Yociss
Reg. No. 36,975
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants